# Erhöhung der Sicherheit von Lebensmittel-warenketten durch Modell-getriebene Prozess-Implementierung

Jens Gulden
Thomas Barth
Dogan Kesdogan
Fatih Karatas

Veröffentlicht in:
Multikonferenz Wirtschaftsinformatik 2012
Tagungsband der MKWI 2012
Hrsg.: Dirk Christian Mattfeld; Susanne Robra-Bissantz


Braunschweig: Institut für Wirtschaftsinformatik, 2012

# Erhöhung der Sicherheit von Lebensmittel-warenketten durch Modell-getriebene Prozess-Implementierung

**Jens Gulden, Thomas Barth, Dogan Kesdogan, Fatih Karatas**[1]

Lehrstuhl für IT-Sicherheitsmanagement, Universität Siegen, Hölderlinstr. 3, 57076 Siegen
{gulden|barth|kesdogan|karatas}@wiwi.uni-siegen.de

## Abstract

Organisationsübergreifende Geschäftsprozesse, wie sie beispielsweise im Rahmen von Lebensmittelwarenketten auftreten, stellen besondere Anforderungen an IT-Lösungen für den Entwurf, die Ausführung und die Nutzungskontrolle der Prozesse. Ein Ansatz besteht darin, Aktivitäten entlang der Warenkette mittels Geschäftsprozessmodellen (business process models, BPMs) zu beschreiben und dabei um Informationen über Sicherheits-anforderungen konzeptionell anzureichern. Diese Modelle können genutzt werden, um eine nach Maßgabe der annotierten Sicherheitsanforderungen implementierte Dokumentation und Nutzungskontrolle transparent und vertrauenswürdig zu realisieren. Wir präsentieren einen Ansatz, der mittels Modelltransformationen eine formale Beziehung zwischen sicherheits-annotierten BPMs und ausführbaren, sicheren Workflows beschreibt.

## 1 IT zur Sicherung von Lebensmittelwarenketten

Im Lauf der Industrialisierung der Nahrungsmittelproduktion ist der Lebensmittelsektor zu einem komplexen, hochgradig dynamischen und wechselseitig verflochtenen Marktsektor geworden. Die Folge ist, dass ablaufende organisationsübergreifende Prozesse gegenwärtig nur schwer kontrolliert und auf unvorhergesehene Einflüsse hin überprüft werden können, wodurch die Prozesse anfällig für Störungen sind. Durch die unüberschaubare Landschaft von unterschiedlichen Marktteilnehmern besteht außerdem ein gesteigertes Risiko intendierter schadhafter Einflüsse auf die Marktprozesse und Lebensmittellieferketten.

Sicherheit auf der Ebene von Warenlieferketten ist mit der „traditionellen“ Sicht der Konzepte und Werkzeuge der IT-Sicherheit nicht effizient zu erreichen, da diese Sicht weitgehend auf die isolierte Sicherung einzelner Systeme beschränkt ist. Da aktuelle Szenarien – zum Beispiel im Bereich Unternehmens-übergreifender Lieferketten und der dazugehörenden

[1] Die in diesem Artikel vorgestellten Forschungsarbeiten wurden in Teilen gefördert durch das Projekt ReSCUe-IT (Robustes und vErfügbares SCM-UntErstützende IT-Plattform), BMBF- Förderkennzeichen 13N10964.

Geschäftsprozesse – dieser Sichtweise nicht entsprechen, besteht eine Lücke zwischen den Konzepten und Werkzeugen zur Sicherung einzelner Systeme und der ablaufenden Geschäftsprozesse [9].

Entgegen traditionellen Fragestellungen der IT-Sicherheit, bei denen die Absicherung einzelner, isolierter Systeme im Vordergrund steht, stellen sich bei der Konzipierung von Sicherheit bei organisationsübergreifenden Geschäftsprozessen neue, bisher wenig bearbeitete Fragen. Beispielsweise sind Geschäftsprozesse, die mehrere Supply Chain Partner betreffen, unweigerlich mit dem Problem konfrontiert, dass auf Geschäftsdaten durch Partner zugegriffen werden muss. In diesem Zusammenhang ist es erforderlich, den Zugriff und die Nutzung von z. T. sensiblen Geschäftsdaten durch Partner zu kontrollieren. Während klassische Zugriffskontrollen lediglich regeln, wer auf welches Datum zugreifen darf, spezifizieren Nutzungskontrollen zusätzlich, welche Aktionen mit den zugegriffenen Daten durchgeführt werden dürfen. Diese Regeln können vor, während oder nach der Ausführung der Aktionen überprüft werden [9]. Die Durchsetzung von Nutzungsregeln sowie das Aufspüren von Verstößen stellen große Herausforderungen im Supply Chain Management dar.

Zum Umgang mit Sicherheitsanforderungen in verteilten Szenarien ist es erforderlich, eine Perspektive auf das Gesamtsystem einnehmen zu können. Konzeptionell erfolgt dies durch die Modellierung organisationsübergreifender Prozesse. Eine übergreifende Perspektive ist außerdem notwendig, um nach Störungen und Angriffen die Funktionsfähigkeit der Warenketten wiederherzustellen, aufgetretene Probleme zu lokalisieren und vorbereitete Kompensationsmaßnahmen einzuleiten. Es gilt, je offener und globaler die Sicht auf den Zustand der Waren und Prozesse ist, desto schneller und effektiver kann auf Störungen und Angriffe reagiert werden. Diese Anforderung der allgemeinen Verfügbarkeit und Integrität steht jedoch im Widerspruch zu den Vertraulichkeits- und Schutzbedürfnissen der einzelnen beteiligten Unternehmen. Diese möchten ihre Geschäftsgeheimnisse (z. B. aktuelle Auftragslage, sensible Kunden- und Zuliefererdaten) durch solch eine Überwachung nicht gefährden. Die technische Realisierung einer übergreifenden Lieferketten-Überwachung muss daher soweit als möglich so gestaltet werden, dass die Unternehmen entlang der Warenkette die Sicherheit und den Schutz ihrer Interessen selbst überprüfen können. Diese Kontrollmöglichkeit wird insgesamt die Transparenz erhöhen und das „vertrauen müssen“ minimieren. In der Literatur ist dieser Ansatz unter dem Stichwort „Mehrseitige Sicherheit“ bekannt [11] [13].

Die Umsetzung mehrseitig sicherer organisationsübergreifender Prozesse stellt eine anspruchsvolle Managementaufgabe dar, die ein umfangreiches Maß an organisatorischer Governance und Konsensbildung unter den beteiligten Geschäftspartnern erfordert. Eine technische Unterstützung dieser Aufgaben kann durch die Überführung konzeptionell spezifizierter Sicherheitsanforderungen in automatisierte Kontrollsysteme geleistet werden. Daraus ergibt sich die Forschungsfrage, wie ein Verfahren zu konstruieren ist, das die Spezifikation mehrseitiger Sicherheitsanforderungen aus einer organisationsübergreifenden Perspektive sicher, vertrauenswürdig und von allen Beteiligten akzeptiert in Software-gestützte Kontroll- und Monitorsysteme überführt.

In diesem Beitrag wird ein Ansatz vorgestellt, der die Sicherung von Geschäftsprozessen in einer umfassenden – ausgehend von annotierten Prozessmodellen bis hin zu ausführbarem Code – und transparenten, im Sinne von nachvollziehbaren und zertifizierbaren, Art

unterstützt. Dies wird durch einen Modell-getriebenen Ansatz geleistet, in dem durch Transformationen eine Folge von Modellen stufenweise konkretisiert und mit Teilprozessen oder Prozessschritten erweitert werden, um eine ablauffähige sichere Umsetzung des Gesamtprozesses zu erreichen. Demonstriert wird dieser Ansatz und der aktuelle Stand der Implementierung anhand eines Szenarios aus dem Bereich der Lebensmittel-Zulieferkette, in dem der Aspekt Sicherheit der Lebensmittel und der Aspekt der IT-Sicherheit miteinander verwoben sind.

## 2 Verwandte Forschungsarbeiten

Die vorgestellte Entwicklungsmethode zur Erreichung eines sicheren Systems entlehnt einige grundlegende Konzepte und Vorgehensweisen aus Ansätzen der modellgetriebenen Softwareentwicklung. Eine Verwandtschaft mit dem Model-Driven-Architecture (MDA, [17]) Ansatz besteht darin, dass das Vorgehen explizit zwischen unterschiedlichen Abstraktionsebenen der konzeptionellen Modellierung und damit verbundenen Anforderungsspezifikationen, sowie technischer Modellierung und Darstellung von Implementierungsdetails differenziert.

Mit Ansätzen zur domänenspezifischen Softwareentwicklung (domain specific software engineering, DSSE [3]) teilt der vorgestellte Ansatz die Verwendung einer domänenspezifischen Modellierungssprache mit spezialisierten Sprachkonzepten zum Ausdrücken von Sicherheitsanforderungen. Auf die Erstellung von Meta-Modellen zur Deklaration der verwendeten Modellierungssprachen wird in der vorgeschlagenen Methode wie bei DSSE zurückgegriffen.

Der Ansatz der Integration von Geschäftsprozessen und Ereignissen in der Zulieferkette ist ebenfalls in den Konzepten des Supply Chain Event Management (SCEM) zu finden (siehe z. B. [10] [12]). Software-technisch ist dabei eine – im Falle von [12] auch Service-orientierte – Integrationsplattform für Geschäftsprozesse als Basis vorgesehen. Konkrete technische Implementierungen sind allerdings nicht dokumentiert. Auch wird bei den bisherigen SCEM-Konzepten das Ereignis als 'typisches' Ereignis in der Lieferkette interpretiert, also zum Beispiel das Ereignis des Unterschreitens eines Mindestlagerbestands oder das Ausbleiben einer Lieferung zum geplanten Zeitpunkt. Im hier präsentierten, stärker Sicherheits-orientierten Konzept ist der Begriff des Ereignisses auch auf den Zusammenhang zu einem Risiko für Güter in der Zulieferkette bezogen.

Das Thema SOA Security ist ebenso wie SOA selbst zunächst ein Konzept, welches innerhalb eines Projekts anhand von Sicherheitsanforderungen konkretisiert werden muss [2]. Aufgrund dessen existieren zwar Leitfäden wie beispielsweise das SOA-Security-Kompendium des BSI [16], die darin enthaltenen Vorschläge müssen jedoch für den Einzelfall angepasst werden. Auf der anderen Seite existiert eine Reihe von Standards, mit denen Einzelaspekte von Sicherheitskonzepten abgedeckt werden (z. B. die WS-Standards des W3C [1]). Diese sind ebenso wie die erwähnten Leitfäden im Rahmen eines konkreten SOA Projekts mithilfe von Tools zu einem Sicherheitskonzept zusammenzufügen.

# 3 Ein Beispiel zur Sicherung einer Lebensmittelwarenkette

Das folgende Szenario präsentiert ein Beispiel für eine modellierte Lebensmittelwarenkette mit mehreren verteilt agierenden Geschäftspartnern: ein Lebensmittelhändler platziert eine Bestellung über eine Menge von Tiramisu Frischei-Desserts bei einem Lebensmittel-Produktionsbetrieb. Dieser bestätigt die Bestellung, produziert das bestellte Gut, und beauftragt einen Logistikdienstleister mit dem Transport zum Besteller. Nach erfolgter Lieferung bestätigt der Besteller den Erhalt der Ware beim Produzenten, und der Prozess der modellierten Lebensmittelwarenkette ist beendet.

In einer konzeptionellen Modellsicht stellt sich diese Lebensmittelwarenkette wie in Bild 1 gezeigt dar. Die Abbildung zeigt einen Ausschnitt eines mit einer domänenspezifischen Modellierungssprache erstellten Lieferkettenmodells [14]. Es handelt sich um ein Geschäftsprozessmodell auf hoher Aggregationsebene. Dargestellt sind drei beteiligte Geschäftspartner als vertikale Swimlanes, darin die von ihnen durchgeführten Aktivitäten als rechteckige Elemente, innerhalb derer Informationsressourcen wie das Bestelldokument (im Beispiel „OrderTiramisu“), oder das zu transportierende physische Gut („Tiramisu“) erscheinen. Die Modellierungssprache enthält domänenspezifische Konstrukte, die es erlauben, Sicherheitsanforderungen auf Ebene des Lieferkettenmodells konzeptionell zu spezifizieren. Die Sicherheitsanforderung an die Verwendung signierter elektronischer Dokumente wird im Beispiel durch das Symbol eines Stifts auf Papier in den Prozessschritten Bestellen („Order“) und Bearbeiten der Bestellung („Dispatch“) dargestellt. Zum anderen enthält das Beispielmodell in den Prozessschritten „Transport“ und „Receive“ das Symbol eines Thermometers vor einem Transportcontainer als Indikator der Spezifikation einer einzuhaltenden Kühltemperatur während des Transports.

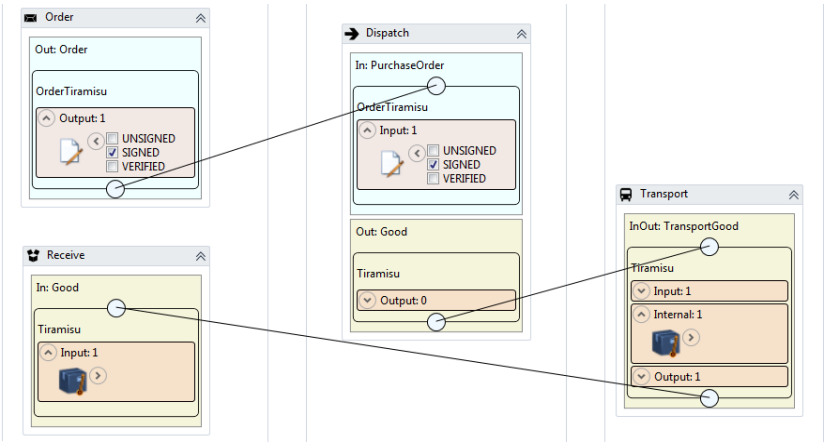


**Bild 1: Ausschnitt eines Lieferkettenmodells in einer mit Sicherheitsattributen angereichten domänenspezifischen Modellierungssprache**

Ausgehend von diesem Modell soll eine dezentrale Plattform realisiert werden, die die Abläufe entlang der modellierten Lieferkette einer sicheren, transparenten Nutzungskontrolle unterzieht. Die Plattform steuert den Nachrichtenaustausch der beteiligten Geschäftsprozesspartner über abgesicherte Verbindungen, wobei zwischen dem Empfangen einer Nachricht und ihrem Weiterreichen unterschiedliche Dokumentations- und Sicherheitsfunktionen entsprechend den spezifizierten Sicherheitsanforderungen im konzeptionellen Lieferkettenmodell aufgerufen werden. Diese Aufgaben der Plattform werden mit der

Workflow-Sprache BPEL [8] [18] realisiert. Bild 2 zeigt einen Ausschnitt der grafischen Visualisierung eines BPEL Prozesses, der durch das vorgestellte Verfahren aus dem obigen Lieferkettenmodell generierten wurde.


**Bild 2: Ausschnitt eines aus dem Lieferkettenmodell generierten BPEL Prozesses**

## 4 Vorgehen

Ziel des Projekts ReSCUe-IT (http://www.sichere-warenketten.de/) ist die Sicherung von Lebensmittelwarenketten gegen beabsichtigte oder unbeabsichtigte Beeinträchtigungen, d. h. gegen gezielte Anschläge oder Naturkatastrophen. Dies soll entlang der gesamten Zulieferkette „from farm to fork“ gelten, also zwischen Herstellung über Transport und Handel bis zum Endkunden. Im Rahmen des Projekts wurden dazu Szenarien identifiziert, in denen eine sichere Softwareumgebung für die Geschäftsprozesse auch die Sicherheit der Güter entlang der Zulieferkette erhöhen kann. Eine 100%-ige Sicherheit der Güter ist allerdings weder durch physische Maßnahmen noch durch IT-gestütztes Vorgehen zu gewährleisten.

Beispielhaft zeigt ein Szenario „Kontamination im Transport“ auf, wie die physische Sicherheit von Lebensmitteln durch Prozess-Governance mit passender Softwareunterstützung gesteigert werden kann: Durch verschlüsselte Übertragung signierter Dokumente (z. B. des Lieferscheins) kann die Integrität der Daten einer Lieferung und die Identität der Absender sichergestellt werden. Eine Manipulation von Produkten bzw. die Verschleierung der Manipulation auf der Basis eines gefälschten Gewichts, einer falschen Anzahl o. ä. kann somit verhindert werden. Durch nicht beobachtbare Kommunikation von Dokumenten entlang der Lieferkette können fundamentale Informationen über Kunde-Lieferant-Beziehungen vor einem möglichen Angreifer (auf die physischen Produkte und/oder die

IT-Infrastruktur der Partner in der Zulieferkette) geheim gehalten werden. Mit Hilfe von an den Transportbehältern der Güter angebrachten Sensoren (z. B. für Licht, Temperatur, Beschleunigung, Luftgüte) ist es möglich, auch während des Transports eine Überwachung des Zustands der Güter zu realisieren. Beispielsweise können bei einzuhaltenden Temperaturgrenzen (z. B. bei Tiefkühlprodukten) Sensoren so konfiguriert werden, dass bei Über-/Unterschreitung automatisch Nachrichten verschickt werden. Über Verfahren des Complex Event Processing (CEP, [4]) können Muster von Ereignissen erkannt und verarbeitet werden, die Messwerte unterschiedlicher Sensoren kombinieren. Auf diese Weise kann zum Beispiel auf ein Kontaminationsrisiko geschlossen werden, das als Abfolge eines steigenden Lichtwerts mit ansteigender Temperatur (möglicherweise durch ein Öffnen des Transportbehälters) und anschließend gemessener Erschütterung des Transportbehälters aufgefasst wird. Die Einbindung solcher Ereignismeldungen in einen übergreifenden Kontrollprozess wird in Abschnitt 5 prototypisch vorgestellt.

Technisch wird die sichere Kommunikationsplattform durch Enterprise Service Bus (ESB)-Implementierungen realisiert. Diese übernehmen sowohl auf der Seite der einzelnen Partner der Zulieferkette als auch auf der Seite der Plattform die Anbindung bestehender IT-Infrastrukturkomponenten mittels abgesicherter Nachrichtenübermittlung. In Bild 3 ist diese Architektur schematisch dargestellt.

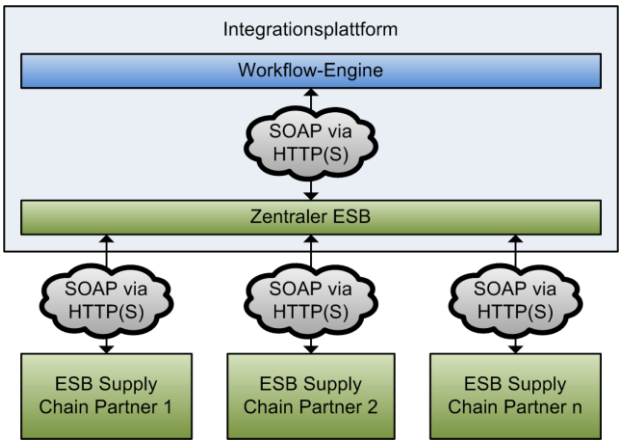


**Bild 3: Verteilte Architektur der sicheren Kommunikationsplattform**

Wie bereits erwähnt, ist die isolierte Betrachtung der Sicherheit einzelner Systeme zur Sicherung eines organisationsübergreifenden Geschäftsprozesses unzureichend. So können beispielsweise auch bei verschlüsselter und digital signierter Übertragung Rückschlüsse auf konkrete Kommunikationspartner gezogen werden, indem Informationen über mehrere Prozessschritte gesammelt und verkettet werden („linkability“ [19]). Die Sicherheit von Daten in diesen Szenarien erfordert also ein systematisches Vorgehen von der Ebene der modellierten Prozesse bis zur Umsetzung der notwendigen technischen Komponenten (Anwendungen, Diensten etc.). Das Vorgehen sollte dabei folgende Komponenten einbeziehen:

1. Die Spezifikation der Sicherheitsanforderungen auf unterschiedlichen Abstraktionsstufen
2. Die Semantik der Sicherheitsanforderungen
3. Transparenz und Nachvollziehbarkeit der Umsetzung
4. Effiziente Werkzeuge zur Spezifikation, Realisierung, Durchsetzung und Überwachung

Zur Umsetzung sicherer Geschäftsprozesse kann eine Vielzahl unterschiedlicher Konzepte (z. B. Verschlüsselung zur Integritätssicherung, Signieren zur Identitätswahrung, Anonymisierung zur Verschleierung der Identität) und entsprechender Werkzeuge herangezogen werden. Die vollständige Beschreibung aller notwendigen Komponenten ist zu komplex, um sie in einem Schritt leisten zu können, insbesondere dann, wenn auf der Ebene der Geschäftsprozessmodelle die eigentliche Domäne mit Fokus auf die Aktivitäten entlang der Zulieferkette im Mittelpunkt steht. Da eine zu komplexe Beschreibung der Sicherheit zu mehr Fehlern in der Spezifikation und damit zu unsicheren Prozessen führt, ist eine Reduktion der Komplexität durch unterschiedliche Abstraktionsstufen (siehe 1) ein Beitrag zur Sicherheit.

Die Angabe von Sicherheitsanforderungen auf einer (im Sinne von 1) höheren Abstraktionsstufe erfordert eine klare Semantik dieser Anforderungen, um eine Modell-getriebene Abbildung auf eine technische Umsetzung zu ermöglichen (siehe 2). Prinzipiell ist dieses Vorgehen bei jeder Realisierung einer nicht-funktionalen Anforderung (welche Sicherheit aus Software-technischer Sicht ist) durch Umsetzung in funktionale Anforderungen notwendig [7].

Vertrauen in eine weitgehend automatisierte Abbildung von abstrakten Sicherheitsanforderungen auf lauffähige Komponenten kann nicht per se vorausgesetzt werden und ist als kritisch für die Akzeptanz des Ansatzes zu sehen. Eine nicht akzeptierte und daher nicht angewendete sichere Lösung kann auch nicht zu sicheren Prozessen führen. Ebenfalls ist eine Zertifizierung als Ausweis der Sicherheit einer Lösung nur denkbar, wenn Vertrauen durch eine nachvollziehbare, d. h. prüfbare, Lösung ersetzt wird. Daher ist Transparenz eine zentrale Eigenschaft des Ansatzes (siehe 3).

Die Dienste der technischen Infrastruktur bilden die Zielplattform, für die die Modelltransformation Code erzeugt. Zur Laufzeit des Prozesses werden diese Dienste durch Infrastrukturkomponenten (Web Server), bzw. individuelle Komponenten, die via Web Service-Schnittstellen verfügbar sind, bereitgestellt (z. B. Services für die Erstellung und Prüfung von digitalen Signaturen). Durch Generierung von Policies kann gewährleistet werden, dass dedizierte Sicherheitsanforderungen beim Aufruf von Diensten erfüllt werden, z. B. die Nutzung von verschlüsselter Kommunikation zwischen Diensten. Die Gesamtheit dieser Werkzeuge unterstützt alle Lebenszyklus-Phasen von Geschäftsprozessen und gewährleistet damit die technisch sichere Ausführung und deren Überwachung (siehe 4).

Der vorgestellte Ansatz verwendet ein Transformationsverfahren vom konzeptionellen BPM zur ausführbaren Software, bei dem Design-Entscheidungen bezüglich der technischen Umsetzung separat in eigenen Modellen erfasst werden. Ein Template-basiertes Code-Generierungsverfahren [5] führt die vorhandenen Modelle zusammen und erzeugt ausführbare BPEL Workflow-Prozessbeschreibungen, die den spezifizierten Abläufen und Sicherheitsanforderungen nachprüfbar genügen. Dieses Vorgehen erlaubt die Umsetzung des gesamtes Spektrums von Nutzungskontrolle, beginnend mit präventiven Mechanismen vor der Ausführung von Prozessen durch Gültigkeitsüberprüfungen der eingesetzten Modelle zur Entwicklungszeit („Modell-Validierung“), über die Dokumentation und Nutzungskontrolle zur Laufzeit, bis hin zu Recovery-Aktivitäten in Reaktion auf aufgetretene Probleme. Als Einordnung in ein generisches Schema zur Charakterisierung von Nutzungskontrolle, angelehnt an [9], zeigt Bild 4 die methodischen Komponenten, die beim vorgestellten Verfahren die Nutzungskontrolle realisieren.

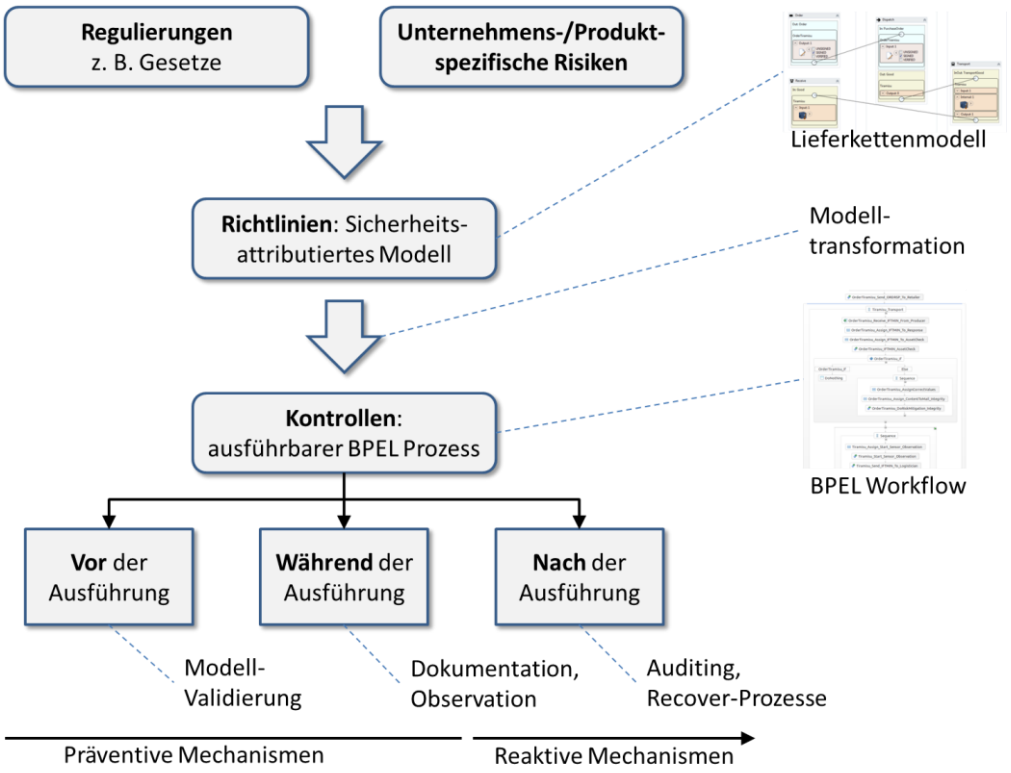


**Bild 4: Komponenten zur Realisierung von Nutzungskontrolle, angelehnt an [9]**

## 5 Prototypische Umsetzung zur Validierung des Verfahrens

Als „proof of concept“ ist das vorgestellte Verfahren implementiert. Der Übergang von konzeptionell modellierten Sicherheitsanforderungen zu automatisiert ausführbaren Kontroll- und Monitoring-Komponenten wird dabei durch Modelltransformationen beschrieben. Der gesamte Transformationsprozess ist in mehrere Schritte untereilt, wodurch unterschiedliche Aspekte des Interpretierens konzeptueller Anforderungen und des Generierens ausführbarer Artefakte methodisch separiert werden. Zunächst kommt eine Modell-zu-Modell Transformation zum Einsatz, die die Konzepte der Lieferkettenmodellinstanz als Eingabe interpretiert und ein Architektur-Modell erzeugt, das technische Implementierungsbeschreibungen zur Umsetzung der konzeptuellen Elemente enthält. Ergänzend wird ein Mapping Modell generiert, das Verknüpfungen zwischen Elementen des konzeptionellen Lieferkettenmodells und den technischen Implementierungskonzepten herstellt. Konfigurationsdetails, wie beispielsweise die Namen von Diensten und deren Endpoint-Adressen, werden aus einer Textdatei in einem Standardformat ausgelesen und in das Architektur-Modell übernommen. Da die Sprachkonstrukte des Architektur-Modells die technischen Komponenten der Zielplattform beschreiben, werden sie Projekt-spezifisch über ein Architektur-Meta-Modell eingeführt. Eine zweite Transformation zur Code-Generierung komplettiert den Übergang vom konzeptionellen Lieferkettenmodell zu lauffähigen Softwarekomponenten und erzeugt ein BPEL-Modell mittels einer Modell-zu-Text Transformation. In Bild 5 sind die beteiligten Komponenten des umgesetzten Verfahrens zusammenfassend dargestellt, einschließlich einer initialen Adapter-Transformation, die das Lieferkettenmodell in ein syntaktisch vereinheitlichtes Format überführt. Die Nummerierung deutet die Reihenfolge an, in der die Transformationen im Lauf des Verfahrens ausgeführt werden.

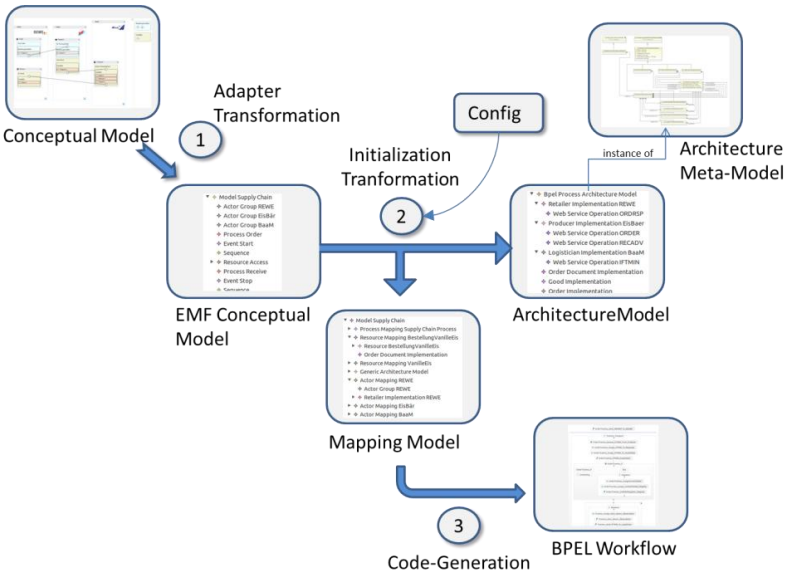


**Bild 5: Komponenten des Verfahrens**

Die Umsetzung der Sicherheitsanforderungen „signierte Bestellung“ und „Sensoren beim Transport“ werden im Folgenden beispielhaft vorgestellt. Die Forderung nach einer signierten Bestellung schlägt sich im konzeptuellen Modell als Annotations-Tag nieder, das dem Prozess-Schritt der Bestellung angehängt ist. Bei der Code-Generierung bewirkt dieses Tag, dass in den erzeugten BPEL Code passende Fragmente eingefügt werden, die in mehreren Schritten und eingebettet in den Gesamtprozess die Eigenschaft der Signierung eines Bestelldokuments überprüfen und je nach Ergebnis den weiteren Verlauf des Prozesses steuern. Bild 6 zeigt einen Teil der dazu verwendeten Transformation in der Sprache Xpand [5].

```
«IF orderReceiveAccess.tags.select( e | e.name=="signatureControl" && e.value.contains("SIGNED") ).is()»
    <!-- Check signature validity -->
    <bpel:assign validate="no" name="«r.name»_AssignValidityCheck">
        <bpel:copy>
            <bpel:from><bpel:literal>«EXPAND Validate_Skeleton»</bpel:literal></bpel:from>
            <bpel:to variable="«r.name»_Validate_Message" part="parameters"></bpel:to>
        </bpel:copy>
        (... copy parameter values ...)
    </bpel:assign>
    <bpel:invoke name="«r.name»_InvokeValidityCheck" partnerLink="«validateService.name»_PL"
        operation="«validateService.operation.operationName»" portType="«validateService.portType»"
        inputVariable="«r.name»_Validate_Message" outputVariable="«r.name»_Validate_Response" />
    <bpel:if name="«r.name»_If">
        <bpel:condition><![CDATA[contains($«r.name»_Validate_Response.parameters/validate:return,string('ok'))]]></bpel:condition>
            <bpel:sequence>
                «EXPAND log(logService, r.name, "ValidityCheckOK", 1, "'electronic signature has been validated OK '")»
                <bpel:empty name="DoNothing" />
            </bpel:sequence>
        <bpel:else>
            <bpel:sequence>
                «EXPAND log(logService, r.name, "ValidityCheckFailed", 1, "'electronic signature validation FAILED'")»
                (... invoke recovery process ...)
        </bpel:else>
    </bpel:if>
«ENDIF»
```

**Bild 6: Einfügen eines Service-Aufrufs in Abhängigkeit von der konzeptuellen Spezifikation**

Eine grafische Darstellung des auf diese Weise in den ausführbaren Prozess übernommenen Implementierungsmusters aus Service-Aufruf und Antwortauswertung zeigt Bild 7. Es sei darauf hingewiesen, dass es sich hier um die prototypische Demonstration der Einbindung eines Services per Code-Generierung handelt, die Umsetzung der konkreten Signatur-Validierung kann in einer endgültigen Fassung an anderer Stelle realisiert sein.

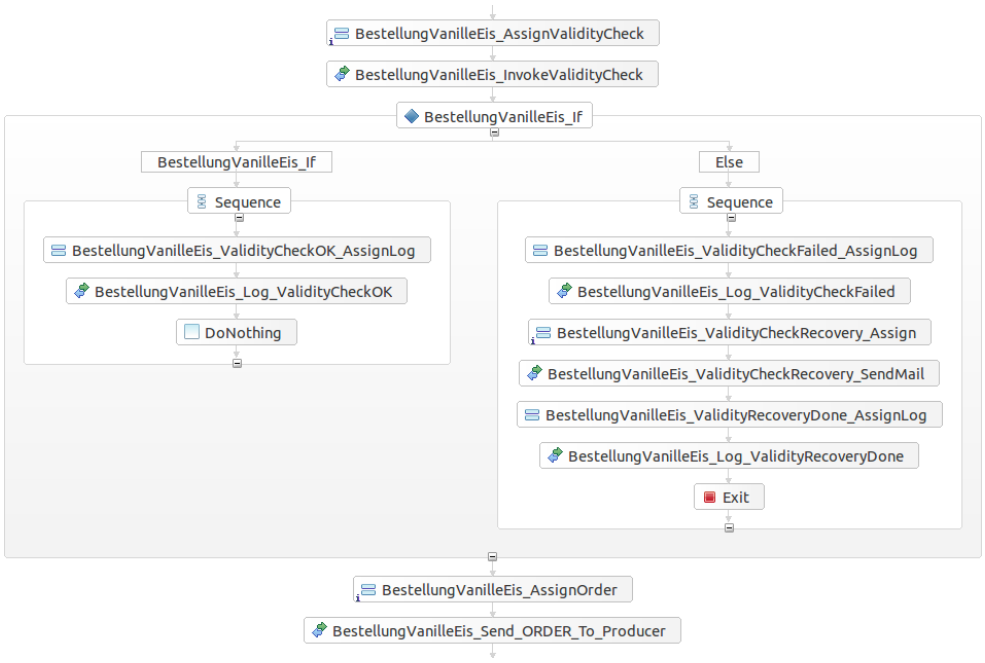


**Bild 7: Grafische Darstellung des in den Prozess übernommenen Implementierungsmusters**

Die Spezifikation der Anforderung zur Überwachung des Transports durch Sensoren geschieht ebenfalls über eine Annotation im konzeptuellen Modell. Diese legt zum Beispiel Parameterwerte für einzuhaltende Temperaturgrenzen fest. Im daraus generierten BPEL Code wird die Sensor-Kontrolle parallel zum Prozess-Monitoring mittels eines Event-Handlers implementiert, der während der Laufzeit des Transports auf Warnungen des Sensor-Monitorings reagiert.

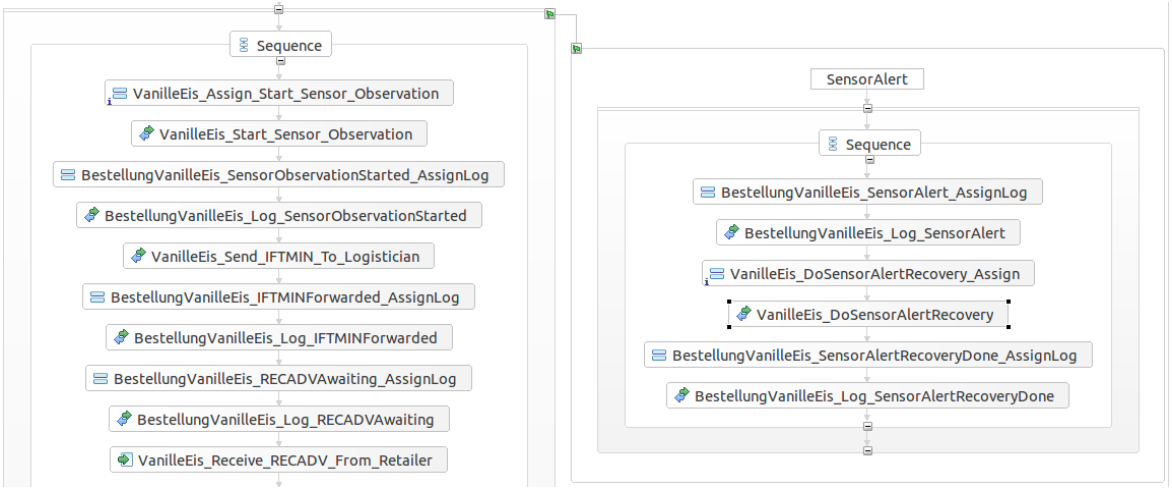


**Bild 8: Generierter Event-Handler Code in grafischer Visualisierung**

In der grafischen Darstellung des BPEL Prozesses erscheint der Event-Handler als paralleler Block zum Code des Supply Chain Prozesses, und ist mit einem kleinen grünen Flaggensymbol in der linken oberen Ecke gekennzeichnet. Bild 8 zeigt den generierten Event-Handler Code in grafischer Visualisierung, rechts neben dem Prozess-Code zur Steuerung des regulären Supply Chain Prozesses. Ein Code-Generator-Fragment, das die Generierung des Event-Handler Codes steuert, ist in Bild 9 skizziert.

```
«LET (! transportProcess.tags.select(e|e.name == "temperatureControl").isEmpty()) AS useSensors»
<bpel:scope name="«rr.name»_Scope">
    «IF useSensors»
    <!-- set up event handler for monitoring sensor data during transport -->
    <bpel:eventHandlers>
        <!-- incoming sensor alert -->
        <bpel:onEvent
            partnerLink="client" operation="«platform.incomingSensorAlert.operationName»" portType="«platform.portType»"
            messageType="«platform.incomingSensorAlert.inputMessageType»" variable="«rr.name»_SensorAlert">
            <bpel:scope>
                <bpel:sequence>
                    «EXPAND log(logService, r.name, "SensorAlert", 3, "'an ALERT from sensor observation has been detected'")»
                    (... invoke recovery process ...)
```

**Bild 9: Generator-Fragment zur Generierung des Event-Handler Codes**

# 6 Zusammenfassung

Im vorliegenden Beitrag wurde ein Konzept vorgestellt, das auf die weitgehend automatisierte Realisierung einer sicheren Prozessausführung für Geschäftsprozesse zielt. Ausgehend von Prozessmodellen, die mit Hilfe von Annotationen um vergleichsweise abstrakte und nicht-technische Aspekte der Sicherheit ergänzt werden, ermöglichen schrittweise Transformationen eine Umsetzung dieser Sicherheitsanforderungen in konkrete Technologien. Beispielhaft wurde dieses Vorgehen an einem Prozess im Rahmen der Lebensmittelwarenkette technisch umgesetzt. Die Architektur der dabei zu Grunde liegenden Service-orientierten Plattform ist dazu bereits in Teilen implementiert, um den hier dargestellten „proof of concept“ zu ermöglichen.

Über den aktuellen Stand der Entwicklung und des Konzepts hinaus sind weitere Themen relevant: Die exakte Formulierung der Abbildung abstrakter Anforderungen (z. B. „sicherer Datenaustausch“) auf Technologien (z. B. Verschlüsselung, Signierung mit bestimmten Verfahren) ist derzeit implizit Bestandteil der Transformation und müsste zur Erhöhung der Transparenz ebenfalls explizit verfügbar sein. Auf der technischen Ebene ist die Sicherung von Anonymität und nicht-beobachtbarer Kommunikation durch Verfahren wie Idemix und TOR eine der nächsten Aufgaben.

# 7 Literatur

[1] Bertino, E; Martino, LD; Paci, F; Squicciarini, AC (2010): Security for Web Services and Service-Oriented Architectures. Springer, Berlin.

[2] Kanneganti, R; Chodavarapu, P (2008): SOA Security. Manning, Greenwich.

[3] Kelly, S; Tolvanen, JP (2008): Domain Specific Modeling: enabling full code-generation. Wiley, Hoboken.

[4] Luckham, D (2002): The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley, Boston.

[5] Stahl, T et al. (2007): Modellgetriebene Softwareentwicklung: Techniken, Engineering, Management. dPunkt, Heidelberg.

[6] Weske, M (2007): Business Process Management: Concepts, Languages, Architectures. Springer, Berlin Heidelberg.

[7] Chung, L, Nixon, BA (1995): Dealing with non-functional requirements: three experimental studies of a process-oriented approach. In: ICSE 1995. ACM, New York.

[8] Mendling, J (2006): Business Process Execution Language for Web Service (BPEL). In: EMISA Forum 26(2): 5-8.

[9] Müller, G et al. (2009): Sichere Nutzungskontrolle für mehr Transparenz in Finanzmärkten. In: Informatik Spektrum, 2010: 3-13.

[10] Otto, A (2003): Supply Chain Event Management: Three Perspectives. In: The International Journal of Logistics Management. Volume 14, Number 2; 2003: 1-13.

[11] Pfitzmann A et al. (1999): Flexible mehrseitige Sicherheit für verteilte Anwendungen. In: Kommunikation in Verteilten Systemen: 132-143.

[12] Wiener, K (2008): Supply Chain Event Management (SCEM): A Strategic Application of Business Process Management (BPM). In: Ijioui , R. et al. (Hrsg.), Strategies and Tactics in Supply Chain Event Management. Springer, Berlin Heidelberg: 215-234.

[13] Wolf, G; Pfitzmann, A (2000): Charakteristika von Schutzzielen und Konsequenzen für Benutzungsschnittstellen. In: Informatik Spektrum 23(3): 173-191.

[14] Monakova, A; Schaad, A (2011): Visualizing security in business processes. In: SACMAT '11 Proceedings of the 16th ACM symposium on Access control models and technologies: 147-148.

[15] Satoh, F; Nakamura, Y; Mukhi, NK; Tatsubori, M; Ono, K (2008): Methodology and Tools for End-to-End SOA Security Configurations. In: O'Conner, L (Hrsg), Proceedings of the 2008 IEEE Congress on Services 2008 – Part I. Honolulu.

[16] Bundesamt für Sicherheit in der Informationstechnik (2009): SOA-Security Kompendium 2.0. https://www.bsi.bund.de/cae/servlet/contentblob/486838/publicationFile/30662/SOA-Security-Kompendium_pdf.pdf. Abgerufen am 19. 12. 2011.

[17] Miller, J; Mukerji, J (2003): MDA Guide Version 1.0.1. Object Management Group. http://www.omg.org/mda/mda_files/MDA_Guide_Version1-0.pdf.
Abgerufen am 19. 12. 2011.

[18] Organization for the Advancement of Structured Information Standards (OASIS) (2007): Web Services Business Process Execution Language Version 2.0.
http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.
Abgerufen am 19. 12. 2011.

[19] Pfitzmann, A; Hansen, M (2010): A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management.
http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf.
Abgerufen am 19. 12. 2011.